**Segunda Evaluación Lenguajes de Programación**

**Febrero 3 de 2010**

**Nombre: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. 10

Indicar la salida de la última expresión   
  
fun f1 (nil,milist) = nil     
   |  f1(x::xs,milist) =   ( map x milist) :: f1(xs, milist );  
val l =[ length, hd, hd o tl ];  
val k = [[ 2,4,6,8,10,12,0 ], [0, 1],[1,2] ] ;  
f1 (l,k);

1. 15

Escribir una función ***combinar*** que dada 2 listas, devuelva una lista de lista de pares ordenados, donde cada par ordenado es una combinación de los elementos de la primera con la segunda lista. Puede utilizar la siguiente función en la resolución:

fun com\_unelemento ( e, x::xs) = (e, x):: com\_unelemento( e, xs)  
    | com\_unelemento ( e, nil) = nil;

Ejemplo:

***combinar ([4,7,3,5], [15,18,3,6]);***

***val it = [[(4, 15), (4, 18), (4, 3), (4, 6)], [(7, 15), (7, 18), (7, 3), (7, 6)], [(3, 15), (3, 18), (3, 3), (3, 6)], [(5, 15), (5, 18), (5, 3), (5, 6)]] : (int \* int) list list***

1. 5

Dada la plantilla de definición de clases, defina una clase en Squeak llamada Hoyo,  que herede de CircleMorph, que tenga una variable de instancia llamada profundidad, y que pertenezca a la categoría 'Morphic-Basic'.

***Object subclass: #NameOfSubclass***

***instanceVariableNames: ''***

***classVariableNames: ''***

***poolDictionaries: ''***

***category: ''***

1. 18
2. Indique las variables de instancia comunes a todos los morphs en Squeak
3. Explique cómo generar movimiento de un morph en Squeak
4. Defina equivalencia por nombre y equivalencia estructural (indicar lenguajes)
5. Defina sobrecarga y polimorfismo; dé un ejemplo de cada una
6. Almacenamiento y pase de parámetros en ML
7. Indique ventajas y desventajas de los dos métodos de recolección de basura
8. 20

Graficar el árbol y escribir la notación prefix y postfix para:

**5\*4-3÷2\*8-9**

1. En Ruby
2. en Smalltalk

Explique para qué sirven los registros de activación y cuál es la información que contienen.

1. 20

Indique las firmas de:

**a)fun g( y ) = ( 1,y) :: ( 2, y) :: nil;**

**b)fun r( x, y) = ( x, (0,x+1),(0,y+1));**

**c)fun h( x) = if hd(x) > 1.5 then print (“true” ) else print (“false”)**

**d)fun j x y = x ^ y ;**

Especifique cuál es polimórfica

1. 12

Dado el siguiente procedimiento:

procedure foo is

a : integer;

b : integer;

procedure bar(x, y) is

begin

put1(x);

b = x - b;

y = y + b;

a = y;

x = b + x;

put2(x);

end bar;

begin

a := 5; b := 3;

bar(a,b);

put3(a); put4(b);

end foo;

La función **put** está numerada para clarificar las salidas de cada llamada a **pu**t. El programa producirá salidas diferentes dependiendo del método de pase de parámetros utilizado. Seleccione el mecanismo de pase de parámetros apropiado (por valor, valor/resultado, referencia) que produzca la salida mostrada. Si no hay un método apropiado indique “ninguno”.

(a) put1(x) put2(x) put3(a) put4(b)

**5 7 5 2**

(b) put1(x) put2(x) put3(a) put4(b)

**5 7 7 5**

(c) put1(x) put2(x) put3(a) put4(b)

**5 7 7 4**

(d) put1(x) put2(x) put3(a) put4(b)

**5 8 8 4**